

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau(43) International Publication Date  
26 September 2002 (26.09.2002)

PCT

(10) International Publication Number  
WO 02/075527 A2(51) International Patent Classification<sup>7</sup>: G06F 9/40

Diego, CA 92126 (US). JACOBS, Paul; 9075 La Jolla Shores Lane, La Jolla, CA 92037 (US).

(21) International Application Number: PCT/US02/08394

(74) Agents: WADSWORTH, Philip, R. et al.; Qualcomm Incorporated, 5775 Morehouse Drive, San Diego, CA 92121-1714 (US).

(22) International Filing Date: 18 March 2002 (18.03.2002)

(25) Filing Language: English

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(26) Publication Language: English

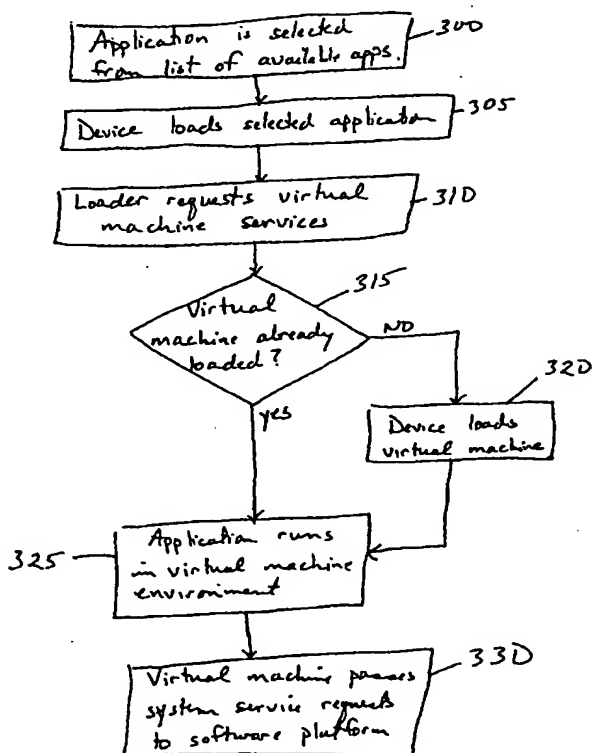
(30) Priority Data:  
60/277,607 19 March 2001 (19.03.2001) US(71) Applicant: QUALCOMM INCORPORATED [US/US];  
5775 Morehouse Drive, San Diego, CA 92121-1714 (US).

(72) Inventors: SPRIGG, Stephen, A.; 121221 Travertine Court, Poway, CA 92064 (US). MINEAR, Brian; 7516 Collins Ranch Terrace, San Diego, CA 92130 (US). MOORTHY, Mahesh; 10750 Dabney Drive #43, San

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent

[Continued on next page]

(54) Title: DYNAMICALLY DOWNLOADING AND EXECUTING SYSTEM SERVICES ON A WIRELESS DEVICE



(57) Abstract: An applet requiring a virtual machine is downloaded to a wireless device. A software platform on the device determines that a virtual machine is used by the applet during execution. Without additional user interaction, the virtual machine is downloaded to the wireless device. This allows those applets requiring a virtual machine to be used with devices that don't have the virtual machine already installed. Further embodiments include the dynamic loading of a system service when an application is being loaded that used the system service. A unique identifier associated with the system service may be used to determine if the system service is loaded.



(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

**Published:**

- *without international search report and to be republished upon receipt of that report*

## **DYNAMICALLY DOWNLOADING AND EXECUTING SYSTEM SERVICES ON A WIRELESS DEVICE**

### **BACKGROUND**

[0001] Typically, system services, including virtual machines, viewers and plug-ins, need to be installed on a device in order to be used by other objects or applications that require them. For example, Java applets run on a device that has a Java virtual machine loaded. Consequently, a device that intends to execute the Java applet will install the Java virtual machine on the device. Typically, because the virtual machine, or other system service, needs to be integrated into the device it is executing on, the installation is performed in advance of trying to download or execute applet or other application wanting to take advantage of the system service.

[0002] Some devices, particularly wireless devices, however, have a constrained environment. Memory, including secondary storage and primary for active programs and data, and processing is more scarce than on other larger computer systems. Consequently, it is advantageous to only download or install some system services, such as virtual machines, on an as needed basis. Unfortunately, on these constrained devices, system services are required to be installed or loaded into memory and take up valuable resources even when not used. Furthermore, users who want to use applications on devices that do not already have the application's supporting system services are hampered or prevented from doing so because the system services were not already installed on the device.

### **SUMMARY OF THE INVENTION**

[0003] The present invention satisfies the shortcomings in the art by providing a system and method for dynamically downloading and installing system services, such as virtual machines, viewers, plug-ins, flash-players, other executable content or data, in a device based on the needs of the application running on the device.

[0004] In one embodiment, the present invention involves downloading a system service onto a wireless device to be used with an application also being downloaded. The system service may be downloaded automatically when the application is downloaded without user intervention.

[0005] In yet another embodiment, the present invention also detects whether a virtual machine is present when an application is loaded into the device's memory for execution and loads and executes the virtual machine, if necessary, so the application may run in the virtual machine's environment.

### BRIEF DESCRIPTION OF THE DRAWINGS

- [0006] Reference will now be made in detail to the presently exemplary and preferred embodiments of the invention as illustrated in the accompanying drawings, in which like reference characters designate like or corresponding parts throughout the several drawings. The nature, objectives and advantages of the present invention will become more apparent to those skilled in the art after considering the following detailed description in connection with the accompanying drawings.
- [0007] The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate presently preferred embodiments of the invention and, together with the general description given above and the detailed description of the preferred embodiments given below, serve to explain the principles of the invention. In the drawings:
- [0008] Figure 1 depicts one exemplary embodiment of the environment in which the present invention may be practiced;
- [0009] Figure 2 depicts the process involved with downloading an application which uses a system service consistent with an exemplary embodiment of the present invention; and
- [0010] Figure 3 depicts the process of loading and executing a virtual machine installed on the wireless device in response to loading an application, which uses the virtual machine.

### DESCRIPTION OF AN EXEMPLARY EMBODIMENT

- [0011] Figure 1 depicts one exemplary embodiment of the environment in which the present invention may be practiced. In this embodiment, the device 100 communicates with the server 105 using a wireless network 125. The device 100 may be a wireless device that may transmit and/or receive data and/or voice. The wireless device 100 may request various pieces of information from the server, including applications 110 and system services 115, such as a virtual machine 120, used by the wireless device.
- [0012] The wireless device 110 may contain a processor and memory, primary and secondary, (not shown) used to store, load and execute the applications 110 and system services 115 downloaded from the server. These applications 110 and system services 115 may also interact with a software platform located on the wireless device used to simplify interfacing with the wireless device, such as by providing generalized calls for device specific resources. Such a software platform is the Binary Runtime Environment for Wireless™ (BREW™) software developed by QUALCOMM, Inc., San Diego, California.

- [0013] It will be recognized by those skilled in the art that the wireless device's 100 architecture may also contain an assortment of other components (either not shown or specifically indicated). Such components include, but are not limited to: a display, speaker, microphone, and buttons allowing alphanumeric and symbol inputs. The wireless device may also contain a battery, multiple storage mechanisms, such as ROMs, RAMs, and flash memory, an operating system and a compilation component to aid in the execution of applications, system services, other executable code and manipulation of data, on the device.
- [0014] While the system services and applications are depicted as being located on the server 105, it will be recognized by those skilled in the art that the applications and software may not be physically located at the server. In this case, the server may make a request for the applications and system services for the wireless device from other systems and then download the requested files or may transfer the request to another system for direct transfer of requested files to the wireless device.
- [0015] As is recognized by those skilled in the art, Fig. 1 is one exemplary environment for the present invention. The device may include other types of systems, including wireless and non-wireless devices. In addition, the device may communicate with the server and other systems by a multiple of network types and communication architectures, including wireless and non-wireless, private and public, switched and non-switched, direct links, and any combination thereof.
- [0016] Figure 2 depicts the process involved with downloading an application which uses a system service consistent with an exemplary embodiment of the present invention. A device, such as the wireless device 100, requests an application from the server 105 (Step 200). This request may have been initiated because a user requests a specific application, such as a game, from the wireless device. The device may be configured such that the applications available to the user are not all resident on the device itself, but a representation of those applications that the user may access.
- [0017] This request, however, may be non-user initiated and may include those tasks for system maintenance and for tasks not involving direct user interaction. In addition, the request may be for types of files other than applications, including data, system services, or other types of information.
- [0018] The device then receives the application from the server and stores it (Step 205). In one embodiment, the server sends the requested application to the device. As stated above, however,

the application may not be physically stored at the server; yet, the server may receive the request and initiate the application download to the device.

[0019] The device then checks the application to determine whether the application uses a system service (Step 210). The application may include a reference, or some identifier, indicating that a systems service is used with the application. This reference may be included with the application or associated with the application in some other manner.

[0001] In one embodiment, the application is a Java applet and requires the use of a Java virtual machine (a system service) to execute. The device may contain a software platform, such as Brew™, described above. The device, using the software platform, determines that a Java virtual machine is used by the applet by checking whether there are any references by the applet to an object class indicating a Java virtual machine. In one embodiment, each object class is represented by a unique 32-bit identifier and this identifier can be used to determine which object classes are referenced.

[0021] The device then determines if the system service is already installed on the device (Step 215). In one embodiment, this is performed by checking the internal tables listing the object classes installed on the device. Using the 32-bit identifier referenced by the downloaded application, the internal tables are checked to determine if the referenced object class is installed, or using the above example, whether the Java virtual machine is already installed on the device.

[0022] If the system service is not installed, then the "no" branch is followed and the system proceeds to download the system service (Step 220). Following the above example, if the Java virtual machine used by the downloaded application is not installed in the device, then the device makes a request to the server to download the Java virtual machine. Additionally, if other system services are needed, they may also be downloaded to the device.

[0023] It will be recognized by those skilled in the art that the downloading of system service(s) used by the downloaded application can be performed without any action by the user. Other than some possible transmission delays or indications, the user may be completely unaware that these downloads are occurring. It may, however, be desirable to inform the user that additional downloads are taking place. This is an implementation preference left to those practicing the present invention.

[0024] It will be further recognized by those skilled in the art that the downloading of a system service may be independent of whether the application was downloaded or not (i.e., the downloading of the system service may be initiated because of applications installed at the factory or otherwise transferred onto the device).

[0025] If the system service is already installed in Step 210 or after it is downloaded in Step 220, then the system service is available for when the application is executed. It will be recognized by those skilled in the art that the system service may be downloaded onto the device but not loaded in the device for execution.

[0026] While the above description incorporates the use of the device making the determination of whether the system services are needed and downloading the system service in response to a request from the device, the invention also embodies the process where the server, or other system, performs the determination as to whether system services are needed and downloads the system service based on whether the device already has the system service installed or not.

[0027] Figure 3 depicts the process of loading and executing a virtual machine installed on the wireless device in response to loading an application which uses the virtual machine. The process begins by having an application selected from those applications available to the wireless device (Step 300). This selection may be performed by a user wishing to execute the application. The selection, however, may be performed without user intervention by the device or in some other automated manner.

[0028] After the application is selected, the device loads the application into memory (Step 305) for execution. (Depending on the environment in which the application is executing, the loading of the application may be considered part of the application's execution). During this loading phase, the loader (the component loading the application for execution) requests the virtual machine services (Step 310). In one embodiment, the loader may perform this using an Applications Programming Interface (API) mechanism built in the software platform, described above, identifying the virtual machine using a unique class identifier.

[0001] For example, the application may be a Java applet requiring the use of a Java virtual machine to execute on the wireless device. A loader loading the Java applet on the device for execution may request Java virtual machine services by using a Brew™ API mechanism identifying the Java virtual machine by a unique identifier. In one embodiment, this identifier is a 32 bit class identifier.

[0030] The device then determines whether the virtual machine system service is loaded into memory (Step 315). The device may do this by checking the object classes loaded into memory. It is preferable that an identifier associated with each object class be used in order to track those system services, applications, executable files, data, other data types or object classes that are loaded.

- [0031] The device, or specifically in one embodiment the Brew software platform, makes the determination as to whether the virtual machine requested by the loader is in memory already.
- [0032] If the virtual machine is not already loaded into memory as determined in Step 315, the "no" branch is followed and the device loads the virtual machine (Step 320). If the virtual machine is not already installed on the wireless device, the virtual machine may be downloaded onto the wireless device from an external source, such as a server or other computer system which has access to the virtual machine software.
- [0033] Depending on the device and operating platform the present invention is implemented on as well as the system service, the system service may require an additional execution or start step following its loading. The system service, or virtual machine in this embodiment, should be in a state in the device that is accessible to the downloaded application when executed or possibly other processes.
- [0034] Following step 320 or if the determination is made after Step 315 that the virtual machine is already loaded, the application then runs in the virtual machine environment (Step 325). If the system service is not a virtual machine, the application can now use the system service downloaded for the application. In the Java virtual machine example, the Java applet executes in the Java virtual machine environment.
- [0035] If the device contains a software platform, such as the Brew™ software, to simplify the interface from the application to the wireless device, the virtual machine passes system services' requests made by the application or the virtual machine to the software platform (Step 330).
- [0036] It will be recognized that while figure 3 discusses an application using a virtual machine during execution, that this is for exemplary purposes and the dynamic loading of other system services, other than the virtual machine, other executable content, and data used with applications, are considered within the scope of the present invention.

## CONCLUSION

- [0037] The present invention allows for the dynamic download and execution of system services on a device. In one embodiment, an applet requiring a virtual machine is downloaded to a wireless device. A software platform on the device determines that a virtual machine is used by the applet during execution. Without additional user interaction, the virtual machine is downloaded to the wireless device. This allows those applets requiring a virtual machine to be used with devices that don't have the virtual machine already installed.



[0038] Another embodiment of the present invention allows the dynamic loading of a system service, such as a virtual machine, when an application is being loaded that uses the system service. In this embodiment, it is performed using a unique identifier associated with the system service that allows the device to determine if the system service is loaded.

[0039] The foregoing description of an implementation of the invention has been presented for purposes of illustration and description. It is not exhaustive and does not limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing of the invention. For example, the described implementation includes software but one embodiment of the present invention may be implemented as a combination of hardware and software or in hardware alone. The invention may be implemented with both object-oriented and non-object-oriented programming systems. Additionally, although aspects of the present invention are described as being stored in memory, those skilled in the art will appreciate that these aspects can also be stored on other types of computer-readable media, such as secondary storage devices, like hard disks, floppy disks, or CD-ROM; a carrier wave from the Internet or other propagation medium; or other forms of RAM or ROM. The scope of the invention is defined by the claims and their equivalents.

**We claim:**

## CLAIMS

- [c1] 1. A method for receiving a virtual machine at a wireless device comprising:  
sending a request to download an application using the virtual machine on the wireless device;  
receiving the application and an associated reference from a server, wherein a reference indicates that the application uses the virtual machine;  
using the reference, determining whether the virtual machine is loaded onto the wireless device; and  
in the event the virtual machine is not loaded onto the wireless device, sending a request to the server and downloading the virtual machine onto the wireless device.
- [c2] 2. The method of claim 1 further comprising receiving a request for the application.
- [c3] 3. The method of claim 1, wherein the application is java code and the virtual machine is a java virtual machine.
- [c4] 4. The method of claim 1, wherein the application uses the virtual machine when executing.
- [c5] 5. A method of processing data having an associated system service, comprising:  
determining which system service is associated with the data;  
determining whether the system service is accessible by the wireless device; and  
in the event the wireless device does not have access to the service, gaining access to the system service via a wireless network.
- [c6] 6. The method of claim 5, wherein the data is executable code.
- [c7] 7. The method of claim 6, wherein the system service is a java virtual machine and the executable code uses the java virtual machine when executing.
- [c8] 8. The method of claim 5, wherein a reference is associated with the data and further comprising determining which system service is associated with the data by checking the reference.

- [c9] 9. The method of claim 5 further comprising receiving the system service into the wireless device via the wireless network.
- [c10] 10. The method of claim 5 further comprising receiving the data into the wireless device via the wireless network.
- [c11] 11. A method of receiving a system service on a wireless device, comprising:  
storing a data object onto the wireless device;  
determining whether the data object has a system service associated with the data object;  
determining whether the system service is located within the wireless device;  
in the event the system service is not located within the wireless device, sending a request to a server for the system service; and  
receiving the system service from the server via a wireless network.
- [c12] 12. The method of claim 11, further comprising receiving a request for the data object at the wireless device.
- [c13] 13. The method of claim 11, wherein determining whether the data object has a system service includes analyzing a reference associated with the data object and containing an indication of the system service.
- [c14] 14. A method of processing an application in a wireless device, comprising:  
receiving a request for an application;  
checking the wireless device to determine whether the application is accessible by the wireless device;  
in the event the application is not accessible, requesting access to the application from a server using a wireless network;  
receiving the application and a reference associated with the application, wherein the reference indicates a system service for use with the application;  
checking the reference and determining whether the system service is accessible by the wireless device; and  
in the event the system service is not accessible by the wireless device, requesting access to the system service from a server using a wireless network

- [c15] 15. The method of claim 14 further comprising receiving the application from a server via a wireless network.
- [c16] 16. The method of claim 14 further comprising receiving the system service from a server via a wireless network.
- [c17] 17. An apparatus for processing an application, comprising:  
a wireless device having a storage and a processor;  
the process configured to determine a system service associated with data, to determine whether the system service is accessible by the wireless device, and in the event the wireless device does not have access to the service, gains access to the system service via a wireless network.
- [c18] 18. A computer readable medium containing computer executable instructions that when executed, perform a method for receiving a virtual machine at a wireless device, comprising:  
sending a request to download an application using the virtual machine on the wireless device;  
receiving the application and an associated reference from a server, wherein a reference indicates that the application uses the virtual machine;  
using the reference, determining whether the virtual machine is loaded onto the wireless device; and  
in the event the virtual machine is not loaded onto the wireless device, sending a request to the server and downloading the virtual machine onto the wireless device.
- [c19] 19. A system to receive a virtual machine at a wireless device, comprising:  
means for sending a request to download an application using the virtual machine on the wireless device;  
means for receiving the application and an associated reference from a server, wherein a reference indicates that the application uses the virtual machine;  
means for using the reference, determining whether the virtual machine is loaded onto the wireless device; and  
means for in the event the virtual machine is not loaded onto the wireless device, sending a request to the server and downloading the virtual machine onto the wireless device.

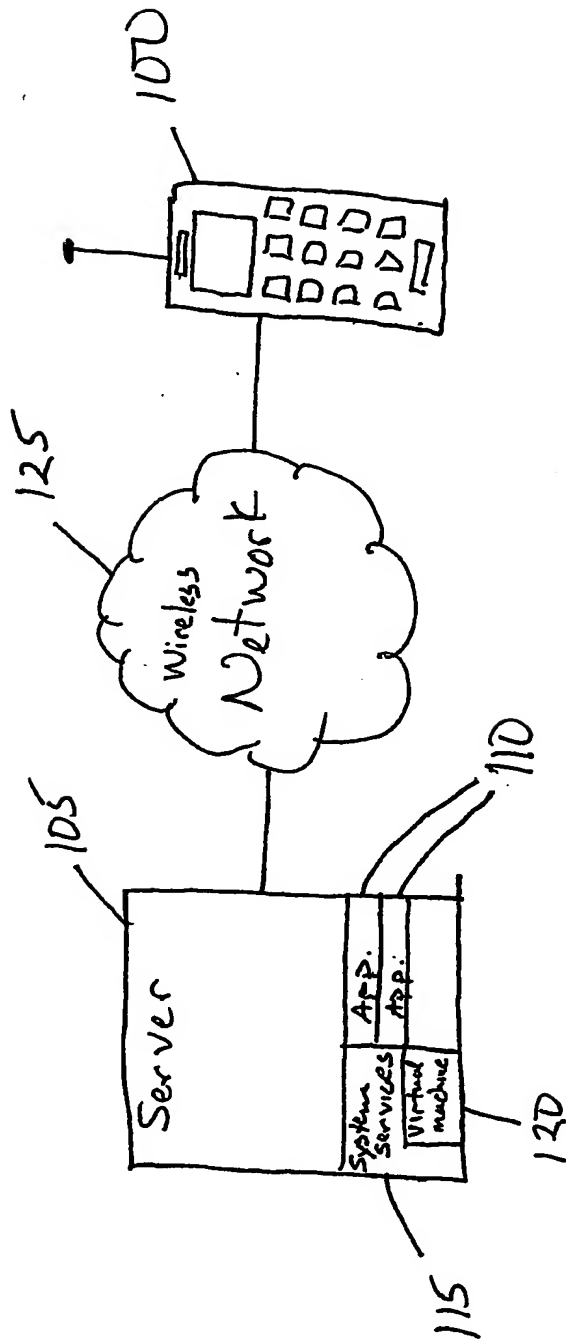


Fig. 1

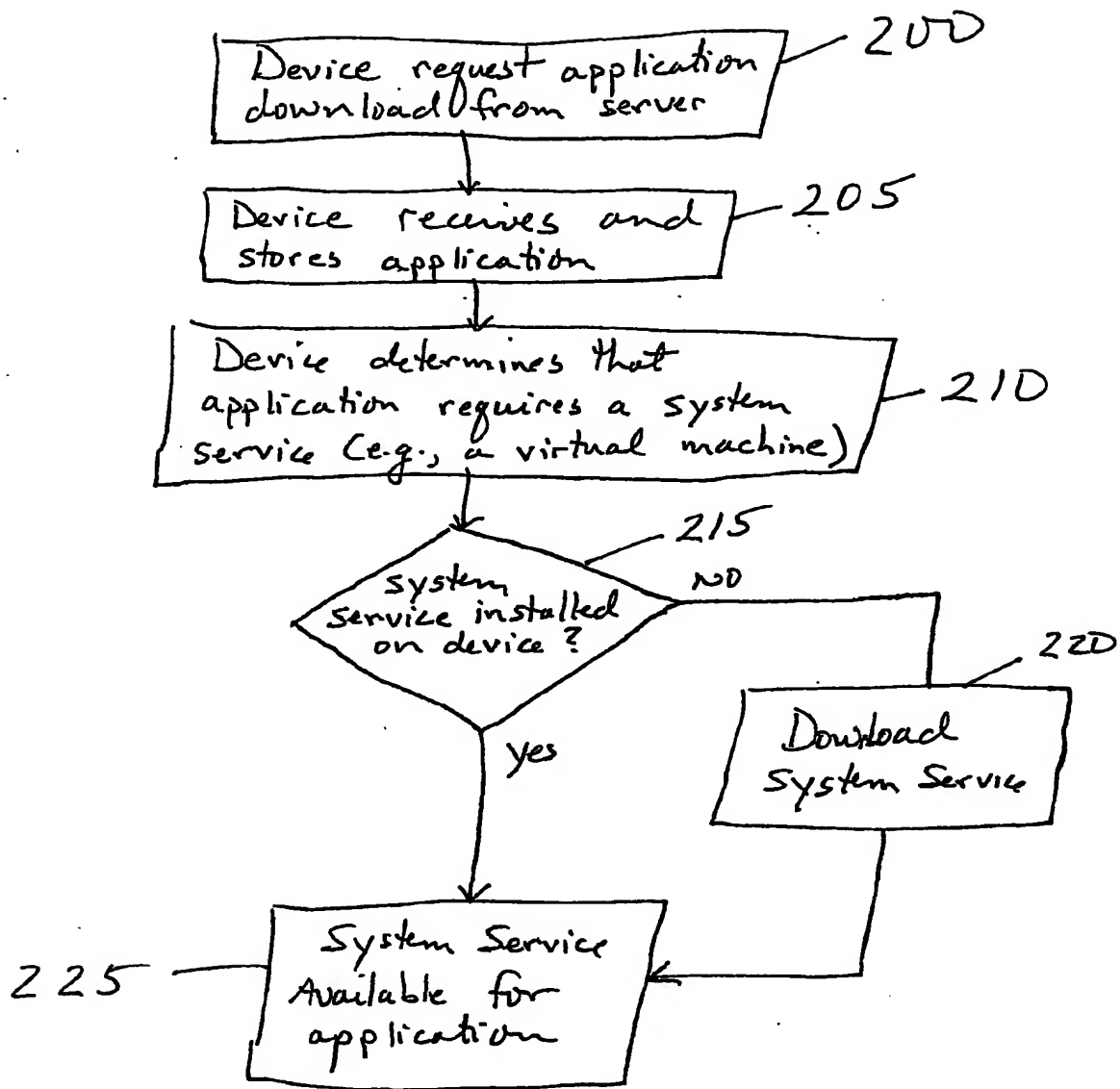


Fig. 2

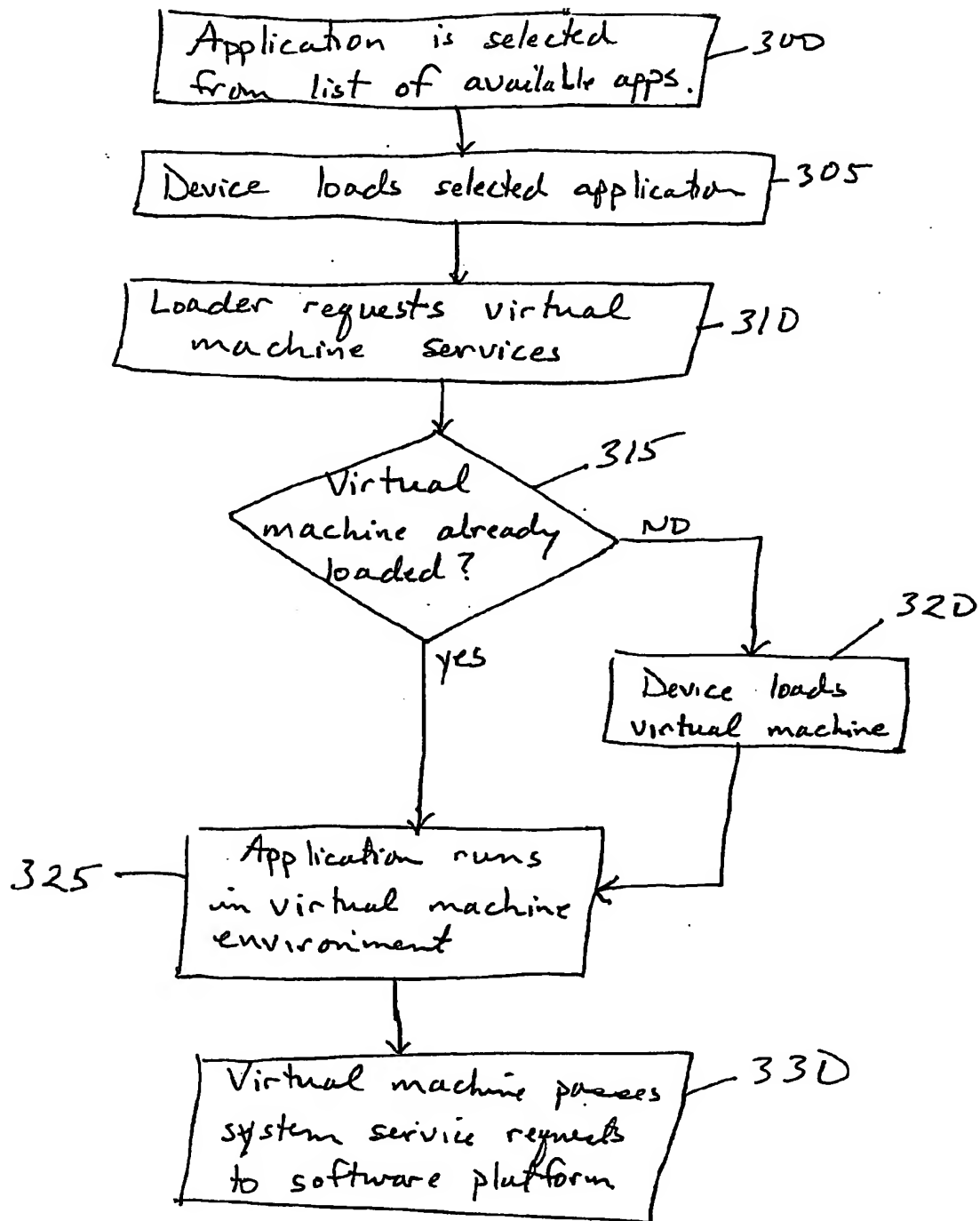


Fig. 3

**This Page Blank (uspio)**